Frontiers in Sensors (FS) Volume 3, 2015 www.seipub.org/fs
doi: 10.14355/fs.2015.03.001

# Advantages of Heterogeneous Agent Populations for Exploration and Pathfinding in Unknown Terrain

Florian Blatt[*1], Matthias Becker[2], Helena Szczerbicka[3]

Simulation and Modelling Group, Faculty of Electrical Engineering and Computer Science
Leibniz University Hannover, Germany

[*1]blatt@sim.uni-hannover.de; [2]xmb@sim.uni-hannover.de; [3]hsz@uni-hannover.de

***Abstract***

In this work we demonstrate how the deployment of several types of agents increases the efficiency and the overall success concerning the task to explore unknown terrain, and finding a pathbetween a starting point and various points of interest. The used agents have different capabilities that are typically foundin technical assistance systems used in search and rescueoperations. In our test cases, the environments to be explored have both, regular characteristics like a maze or a building as well as irregular structures.Our simulations using heterogeneous and cooperating agent populationsshow, that this approach is superior to homogeneous populations, with a higher rate of finding the destinations and in shorter time.

The results should be applicable for strategies in emergency incidents, and search and rescue operations, such as the robot-aided search for victims after an earthquake or other disasters, where formerly known terrain would be inaccessible forhuman rescue helpers.

***Keywords***

*Search and Rescue; Multi-Agent Systems; Ant Algorithms*

## Introduction

At the moment, robot-aided search and rescue operationsmostly rely on robots operated by a human controller, while autonomous robots are expected to play a bigger role in the future. Prototypes of autonomous robots for search and rescue operations already exist (Ruangpayoongsk et al. 2005, Birk et al. 2006, Nagatani et al. 2011). When robots have to act completely autonomously in an unknown area,a strategy has to be implemented how to explore the area most efficientlyand how to signal the results of the exploration phase back to human rescue teams.First approaches use one type of robots only, while some algorithms exist, that use different types of robots or agents for search and rescue scenarios (Kitano et al. 1999, Ferranti et al. 2007, 2009, Zhu & Wang 2008). In this work we review the strategies found in the literature. In our simulations we demonstrate the advantages of our approach using several types of cooperating agents compared to the strategies found in literature.

The typical search and rescue scenario in this case is the following: therescue forces arrive at the site of the disaster and can not enter the regionwithout endangering themselves or the region is unaccessible for anotherreason. One of the most recent real world scenarios would be the disaster at theFukushima Daiichi Nuclear Power Station (Nagatani et al. 2011).

In order to make the simulation results applicable for real scenarios, the agents have tocomply to some limitations concerning their abilities of movement, sensing and communication. These limitations result from the characteristics of available hardware sensors: gyroscope, laser range finder, cam, sonar,infrared, bumper sensor, compass and a laser scanner (Ruangpayoongsak et al. 2005, Birk et al. 2006).

A multi-agent algorithm dependson the communication between its agents. Two types ofinformation transfer can be used: direct and indirectcommunication. The direct variant is facilitated through the use of wireless radio. Indirect exchange of data can be achieved through the use of markingsin the area. These marking can be color marks or the agents could drop RFID chips (Ferranti et al. 2009). Wireless transmission is restricted in its range, so that exchanging data directly with a central instance can not be realized.However the possibility of establishing a transmission path using an ad-hoc routing protocol should be possible.

In the simulations the terrain is given as a matrix representing the coordinates of the region. Thesize of the cells depends on the sensor range and the

communication range of the agents. Each entry of the matrix containsinformation about the terrain, whether it is passable or not, and can holdmarking information left behind by an agent that has visited this cell already. A cell can also contain a point of interest, representing an exit, a danger or a victim. The agents have only local information available or information, that is shared by other agents.

Considering these limitations of a typical search and rescue scenario, wepropose several types of agents, which in turn will be combined to variousalgorithms, that should be able to explore maze-like unknown environments. Whiledoing this they should fulfill the following goals, with descending priority:

1. Explore the region
2. Find points of interest and mark them
3. Mark paths to the points of interest and share them
4. Optimize these paths

## State of the Art

As mentioned above, some algorithms exist already, that use the multi-agentmodel in connection with the search and rescue scenario. The first algorithm isthe typical ant algorithm. This method uses simple agents, that communicate viaindirect communication (Wagner et al. 1999, Svennebring & Koenig 2004, Ferranti et al. 2007). It isnot very effective, depending on the target map, as most of the agents will stayin already explored parts of the region, if the map contains a lot of obstaclesor resembles a maze.

The second existing algorithm is the *Multiple Depth First Search* (DFS) (Ferranti et al. 2007), which simulates the well known depth first search usedon graphs instead on a matrix. It has the drawback, that it has to visit eachcell that is marked as way at least twice.

The *Brick & Mortar* algorithm from Ferranti, Trigoni and Levene also uses indirectcommunication, so that the different agents know, which parts of the region already have been visited and need notbe visited again (Ferranti et al. 2007). A similar algorithm is *CLEAN*, which seems to bethe predecessor (Wagner & Bruckstein 1995).

*HybridExploration* is another approach by Ferranti, Trigoni and Levene (Ferranti et al. 2009).This algorithm extends the previously mentioned*Brick & Mortar*algorithm with stationary sensor nodes. These stationarynodes allow the mobile nodes, the agents, to speed up the explorationprocess. The agents are able to compute some *virtual agents*, that use thedeployed stationary nodes as a network to compute loops, thus allowing theagents to save time at the costs of computing power and the ability to carrysensor nodes.

Howard, Parker and Sukhatme present an algorithm that uses a heterogeneous team of robots toexplore an unknown building and/or region and notify the human controller of itsfindings (Howard et al. 2006). This method uses a frontier-basedalgorithm on all its agents in the exploration phase. A central control forvarious coordination steps is still needed, which is in turn the drawback ofthis algorithm, as it can not run autonomously.

Most of algorithms above (except (Howard et al. 2006)) share thecharacteristic, that they only use a single population of agents tocompute their goal. The algorithm from Howard et al. uses a heterogeneous multi-agent model, however with the drawback, that it depends on the human controller and is not fullyautonomous.Since in previous works, no multi-agent approach without central/human control is found, we developed our approach usingheterogeneous populations without a central control.

## Basic Types of Agents

Our approach uses heterogeneous populations of agents.In this section we introduce the different basic types of agents/robots, which will then be used to create the particular populations.Each type of agent uses a different model of movement. Thesimplest type of agent uses a random movement model most of the time. Others tryto follow walls or orient themselves on markings left behind by otheragents.

Every agent used in the following simulation experiments is based on a standardagent, that represents an ant-like agent (Wagner et al. 1999, Koenig et al. 2001, Dorigo & Gambadella 1997). Asmentioned earlier, this agent explores the environment randomly. Additionallyit has the ability to leave traces or markings behind, usually calledpheromones, akin to the biological pheromones used by real ants. Other agentscan perceive these markings and alter their movement according to theconcentration of these traces. Analogous to the biological model, the standardagent is able to convey different meanings with different markings. For exampleone marking could inform the agent of imminent danger or point the agent intothe direction of a point of interest.

*Standard Ant Agent*

The standard ant agent used in this experiments is able to place anddifferentiate between two kinds of pheromones. The first one is used to mark theway to the starting point and it is dispersed as the agent is searching for the “food” or the point of interest. This pheromone is called the *home trail*. If an ant agent has found a source of food, it will switch the type ofpheromone, that will get released into the environment. The second type ofpheromone, that the agent will use in this case will point the way towards the “food” source. Hence it is named *food trail*. Using these two types ofmarkings allows the ant agent to gather more information about the surroundingenvironment and allows it to make smarter movement decisions.

In order to use the two trail markings and to “see” the surrounding terraineach agent is equipped with three sensors. They are arranged as follows: One inthe front, one in the front left, and one in the front right. The sensors scanthe area one cell ahead of the agent and allow the ant agent to see, if there isa wall or free way in any of the three inspected cells. If the scanned cell ismarked as passable way, then the possible pheromone concentration on this cellis also measured. The agent's movement algorithm uses the concentration of thetrail markings as input to decide in which direction it should move next. Thehigher the concentration, the higher is the possibility, that the agent willchoose the cell containing this amount of pheromone. Although the algorithm willalways leave at least a very small chance to pick a completely random direction,ignoring any markings in the scanned cells, also compare the five simple rulesof ant path planning from (Parunak 1997).

Combining this movement model and the use of the two available pheromone trailsallows this agent to be able to solve most exploration experiments. The drawbackis, that the results are not very satisfactory, as the exploration of unknowncells happens at random. The major drawback is, that this type of agent has thepreference to follow an already known path. Raising the random movement chancementioned earlier in this movement model would help to create a greaterdeviation from the already explored path, but it would decrease the advantage ofthe two pheromone trails. As a consequence this type ofagent is quite usablefor the optimization of already known paths. However the goal is to explore theunknown region as fast as possible. For this reason the following agents havemore sophisticated movement patterns and models, that try to improve theexploration efficiency of this type of agent.

*Wall Follower Agent*

The Wall Follower agent, and all following agents, are based on the Standard Antagent. Meaning that they are also able to detect pheromone trails and leave thembehind. Only the movement algorithm was adapted, providing other advantages andmaybe also disadvantages.

This type of agent implements a traditional maze solving algorithm. The movementmodel, that this agent uses tries to find the nearest wall and after finding it,will follow this wall. There are two variants of this scheme available, thefirst variant follows a wall on the right hand of the agent, while the secondalternative will follow a wall on the left hand of the agent.

The Wall Follower algorithm will always find a way through a maze, as long asall parts of the walls are connected. If the target is in a part of thelabyrinth, where the walls are not connected to the rest of the maze, then thismethod will not be able to find the way. The Pledge algorithm (Abelson & DiSessa 1986) can be used to overcome this fault. Another possibleway to counter this problem is the addition of a random chance to leave thewall, that the agent currently follows. This random possibility does not need tobe very big and our experiments have shown that a possibility of 0.5% is quitehigh enough for the agent to be able to enter or leave the disconnected partsof the maze.

As mentioned above, this agent is also able to dispense a pheromone trail. Acombination of this type of agent with the Standard Ant agent works well, as theWall Follower will find a first path through the terrain and the ants are ableto use the distributed trail markings to find their own way and, in the end, tooptimize this path.

*Marking Agents*

This group of agents was inspired by the work of Zhu and Wang (Zhu & Wang 2008). The aforementioned method proposed the use of a globallyavailable list, that contains the coordinates of already visited cells. Toaccess this kind of list from any point in the unknown terrain, the agents wouldhave to have the ability to communicate with a central server structure or theywould have to be able to create a

communication network to exchange and updatethis list. As this approach is not really feasible in the proposed scenarios wehad to adapt the idea to our requirements, thus converting the list to markingsin the cells. This means, that this agent will mark an already visitedcell. Other Marking agents are now able to identify this mark, while they usetheir sensors to decide their new movement direction. When an agent encounters acell marked as already visited, it tries to avoid to move to this cell. If theagent has only marked cells in its vicinity, it will move to a random cell.

Similar to the Wall Follower, this agent works well in combination with theStandard Ant agent. Our experiments yielded interesting results for acombination consisting of a small percentage of this kind of agent, a small partcomposed of the Wall Follower agent, and a bigger part of the Standard Antagent. This combination has the advantage, that the Wall Follower and theMarking agents are able to scout the unknown environment and lay trails ofpheromones for the simple ants to follow.

## Experimental Simulation Setup

### *Setup of Heterogeneous Populations*

For our experiments we have chosen three different combinations of agentpopulations. Each combination was tested in different test scenarios. The firstcombination of agents contains only the Standard Ant agent as a benchmark, sothat the results of the other populations can be compared. The secondcombination consists only of Marking agents, while the third, as mentionedearlier, is a combination of all three agents proposed in this paper.

The Standard Ant algorithm includes 100 Standard Ant agents, the Marking Antalgorithm uses 100 Marking agents. The third algorithm, the Cooperating Antalgorithm, consists of ten Wall Follower agents, 20 Marking agents and 70standard ants.

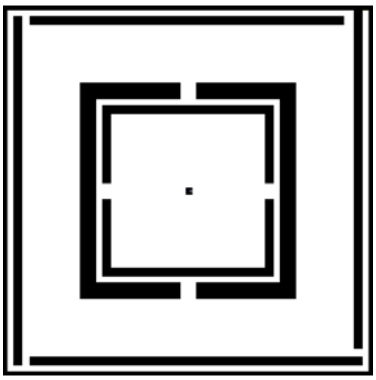

FIG. 1 MAZE 1

FIG. 2 MAZE 2

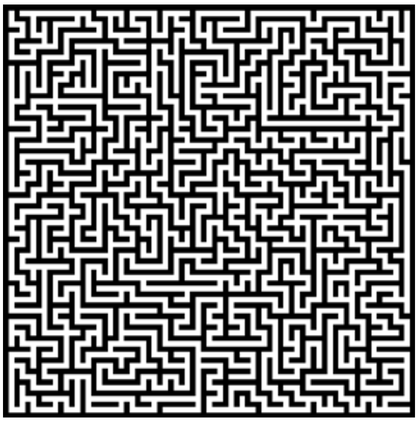

FIG. 3 MAZE 3

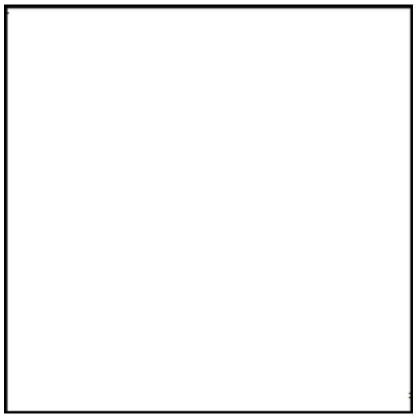

FIG. 4 MAZE 4

FIG. 5 MAZE 5

***Test Arrangements***

Five different maps were chosen to be used in this experiment. All of theseterrains represent different kind of mazes. These differences range from an openplane to a labyrinth generated by a randomized version of the Depth-first searchalgorithm. Some of these mazes may contain unconnected parts of obstacles,resulting in so called islands. Usually the entry point is in the upper left ofthe maps and the exit point in the upper right. There are two exceptions: Maze 1and Maze 4. The entry point in Maze 1 is in the middle of the map, while theexit point of Maze 4 is in the lower right instead of the upper right of themap.

The first goal of each algorithm is toexplore the unknown terrain. The second goal is to find points of interest, which in this this case is the exit point. If the exit point is found, the algorithms will mark apath from the entry to the exit point. Finally each algorithm will try tooptimize this first found path until the algorithm terminates.

The main difference of the five divergent mazes used in this experiment are thethickness of walls and the path width. Some samples have very narrow path waysavailable with broad walls, while others offer extensive open areas with thin oralmost no walls. All examples were created using the GNU Image ManipulatingProgram. Figure 4 was generated through the random Depth-firstalgorithm, Figures 1 to 3 and 5 were drawn by hand.

***Simulation Experiments***

Time in this experiment is measured in simulated ticks. Each experiment is terminated after specific amount of time, depending on the size of the test region: runtime is 100,000 ticks for a maze of size 500x500 cells,120,000 ticks for size of 750x750 cells and 150,000 for a map of size of 1000x1000.Each agent moves fromits current position to a new cell according to its own movement model everytick.

Because each algorithm has a stochastic component, eachsetup has been simulated 25 times. The following diagrams (Fig. 6-10) contain themean value of these 25 runs as well as the confidence interval (for 95% confidence level) for each algorithm and test case.

We defined the length of a path as the sum of all cells, that are needed tocreate a connection between the entry and the exit point. To calculate this sum,all eight cells around the a single cell in the matrix were used as availablepaths.

For a better comparison we also included the results of the *Brick and Mortar* in Table 1. These values will be compared with the timeit took the various algorithms to find the first path, as this method does nottry to find a path and optimize it.

## Results and Discussion

The diagrams in Figures 6 to10represent the results of our simulation experiments. Each diagram depicts theoutcome for all three algorithms applied to the specified maze. The X-axis showsthe different methods, while the Y-axis delineates two different bars. The leftbar of each algorithm, colored blue, shows the length of the first found path incells. The second bar, colored red, portrays the final, optimized path, thateach algorithm has found at the time of the termination of the algorithm. Thisoptimized path is the result of continuous improvement through the agents. Ifno result could be found in any of the 25 simulation runs for onealgorithm, then no bar is shown in the respective diagram, compare Figure 8and 10. Each bar includes the 95%confidence interval at the top, which may be to small to be seen on some bars.

The outcome of the various experiments, including the results from the*Brick and Mortar* algorithm, imply, that the more open space is availablefor the agents, the faster the points of interest will be found and a waydiscovered. These more accessible areas allow the different agents to spreadfaster through the whole map, increasing the chance to find the points ofinterest. In contrast the test case with the most convoluted paths, hinderingthe movement of the agents, shows that only one algorithm was able to find away through the maze. The path lengths depicted in the various diagrams confirmthis behavior of the different methods, as the results are nearly identical.

The results of Maze 3 and Maze 5, on the other hand, clearly show, that narrowpathways increase the needed time to find a valuable solution. Table 1 and the diagrams in Figures 8 and 10 show, that the Standard Ant algorithm was unable to find away through the unknown terrain in both cases. The structure of Maze 3 hinderedthe Marking Ant algorithm enough, that it was impossible to find a solution forthis test case with this algorithm in all the 25 simulation runs, while theBrick and Mortar algorithm did find a solution, it took nearly a million ticksto do so. Comparing this outcome with the resulting mean value of theCooperating Ant algorithm

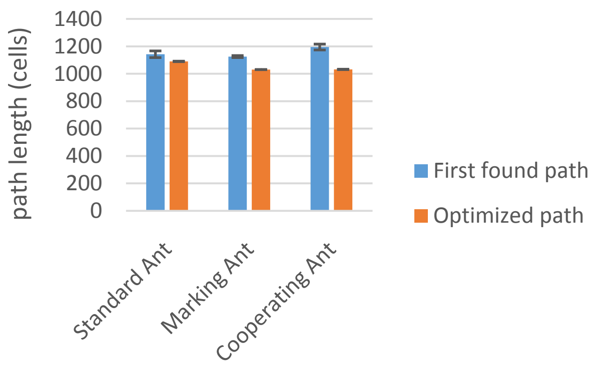


FIG. 6 RESULTS MAZE 1

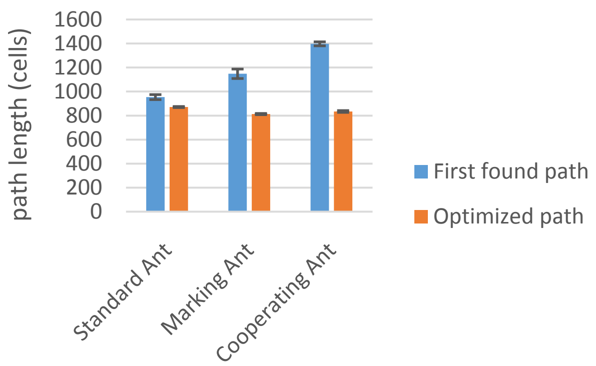


FIG. 7 RESULTS MAZE 2

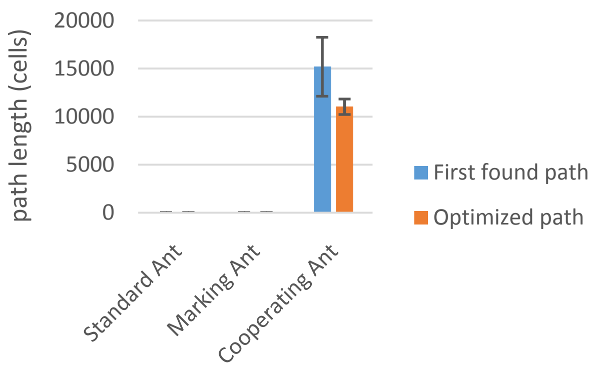


FIG. 8 RESULTS MAZE 3

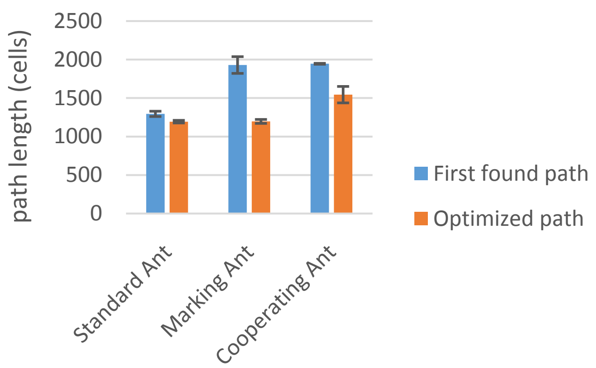


FIG. 9 RESULTS MAZE 4

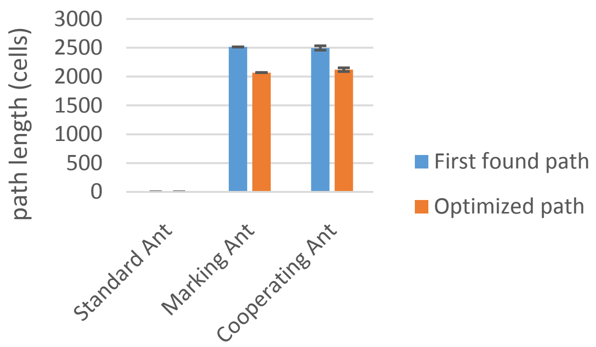


FIG. 10 RESULTS MAZE 5

TABLE 1 STEPS UNTIL THE FIRST PATH WAS FOUND FOR THE DIFFERENT ALGORITHMS

| Maze | Standard Ant | Marking Ant | Cooperating Ant | Brick & Mortar |
|---|---|---|---|---|
| 1 | 44474 | 10283 | 16452 | 65590 |
| 2 | 18816 | 8600 | 2959 | 8660 |
| 3 | - | - | 36559 | 933584 |
| 4 | 39350 | 12734 | 4046 | 2343 |
| 5 | - | 38100 | 77636 | - |
| **Mean** | 34213 | 17429 | 27530 | 25244 |

shows, that the approach of Ferranti et al. method isnot very efficient in this case, as the Cooperating Ant algorithm only needed3% of the time. Our experiments revealed another flaw of the Brick and Mortar algorithm in Maze 5. Figure 5shows a heterogeneous maze, withpredominately narrow paths, which are rather ragged. These ragged paths wouldtrap the agents used by the *B&M* algorithm, culminating in the situation,that all 100 agents trap themselves in these “spikes”.

As the values in Table 1 show, only the Cooperating Ant algorithmwas able to compute results for all five test cases. Compared to the othersolutions, the length of the found paths this algorithm provides are slightlyinferior to the other solutions. The Standard Ant algorithm usually found ashorter path at the first try, but failed to optimize it to the same degree theother two methods did. This is especially evident in Maze 4, the open area testcase. The first found path of the Cooperating Ant algorithm was 2% longer, butat the termination of the simulation, the optimized path was 34% shorter thanthe end result of the Standard Ant algorithm.

The results in Table 1 also show, that the Marking Ant algorithmhas the lowest mean time of all simulated algorithms, even the *Brick andMortar* algorithm, to find the first path. Although the resulting mean time ofthe Cooperating Ant algorithm is influenced through the values of thecomputation of Maze 5, the results for

*B&M* method are dominated from theoutcome of Maze 3.

## Conclusion

In this paper, we examined the possibilities and capabilities of the use ofheterogeneous populations of agents for the exploration of unknown terrain andthe search of points of interest. Our simulation experiments show, that existing algorithms, that rely on a homogeneous population, show variousdeficiencies. These disadvantages are the reason, that these algorithms aresometimes unable to find a solution for this problem. From the four testedalgorithms, only our approach using a heterogeneous populationwas able to compute satisfactory results for all testcases. This algorithm was the only one, which uses all three types of agentsintroduced in this paper.

The results of our experiments clearly indicate, that heterogeneous populations of robots should be used in search and rescue scenarios. Since existing types of robots usually have some kind of communication facility, the realization of the concept of cooperating different kinds of robots is only a little step. It requires mostly the software implementation of different strategies and communications.

Depending on the development of the hardware, our approaches can be optimized in aspects such as number of used agents in the variousalgorithms or as the use of available communication abilities ofagents. All types of agents proposed in this paper use markings on the floor tocommunicate with each other. Additional avenues of communication would alsoincrease the capabilities of cooperation between the various agents.

## REFERENCES

Abelson, H., and A. DiSessa, "Turtle geometry: The computer as a mediumfor exploring mathematics". The MIT Press, 1986.

Birk, A., K. Pathak, S. Schwertfeger, and W. Chonnaparamutt, "The IUB rugbot: an intelligent, rugged mobile robot for search and rescueoperations," in IEEE International Workshop on Safety, Security, andRescue Robotics (SSRR). IEEE Press, 2006.

Dorigo, M., and L. Gambardella, "Ant colony system: A cooperativelearning approach to the traveling salesman problem," IEEE Transactionson Evolutionary Computation, vol. 1, no. 1, pp. 53–66, 1997.

Ferranti, E., N. Trigoni, and M. Levene, "Brick &Mortar: An on-linemulti-agent exploration algorithm," in IEEE International Conferenceon Robotics and Automation. IEEE, 2007, pp. 761–767.

Ferranti, E., N. Trigoni, and M. Levene, "Rapid exploration of unknownareas through dynamic deployment of mobile and stationary sensornodes," Autonomous Agents and Multi-Agent Systems, vol. 19, no. 2,pp. 210–243, 2009.

Howard, A., L. Parker, and G. Sukhatme, "Experiments with a largeheterogeneous mobile robot team: Exploration, mapping, deploymentand detection," The International Journal of Robotics Research, vol. 25,no. 5-6, pp. 431–447, 2006.

Kitano, H., S. Tadokoro, I. Noda, H. Matsubara, T. Takahashi, A. Shinjou,and S. Shimada, "Robocup rescue: Search and rescue in largescaledisasters as a domain for autonomous agents research," in IEEESMC'99 Conference Proceedings. 1999 IEEE International Conferenceon Systems, Man, and Cybernetics, vol. 6. IEEE, 1999, pp. 739–743.

Koenig, S., B. Szymanski, and Y. Liu, "Efficient and inefficient antcoverage methods," Annals of Mathematics and Artificial Intelligence,vol. 31, no. 1, pp. 41–76, 2001.

Nagatani, K., S. Kiribayashi, Y. Okada, S. Tadokoro, T. Nishimura,T. Yoshida, E. Koyanagi, and Y. Hada, "Redesign of rescue mobilerobot quince," in IEEE International Symposium on Safety, Security,and Rescue Robotics (SSRR). IEEE, 2011, pp. 13–18.

Parunak, H., ""Go to the ant": Engineering principles from natural multiagentsystems," Annals of Operations Research, vol. 75, pp. 69–102,1997.

Ruangpayoongsak, N., H. Roth, and J. Chudoba, "Mobile robots forsearch and rescue," in IEEE International Safety, Security and RescueRobotics, Workshop. IEEE, 2005, pp. 212–217.

Svennebring, J. and S. Koenig, "Building terrain-covering ant robots:A feasibility study," Autonomous Robots, vol. 16, no. 3, pp. 313–332,2004.

Wagner, I. and A. Bruckstein, "Cooperative cleaners: A study in antrobotics,"1995.

Wagner, I., M. Lindenbaum, and A. Bruckstein, "Distributed coveringby ant-robots using evaporating traces," IEEE Transactions on Roboticsand Automation, vol. 15, no. 5,

pp. 918–933, 1999.

Zhu, Q. and L. Wang, "A new algorithm for robot path planning basedon scout ant cooperation," in ICNC'08. Fourth International Conferenceon Natural Computation, vol. 7. IEEE, 2008, pp. 444–449.

**Florian Blatt** is a Ph.D. candidate at the Department of Simulation and Modelling - Institute of SystemsEngineering at the Gottfried Wilhelm Leibniz Universität Hannover. He received his Master of Science in Computer Science from the same university. His research interests are organic computing, multi-agent systems, autonomous search & rescue algorithms and exploration algorithms.

**Matthias Becker** got his diploma in computerscience from University Würzburg, Germany in 1996 and hisPh.D. from University Bremen, Germany in 2000. Since then he is researcher and lecturer at the Gottfried Wilhelm Leibniz University Hannover where he got his habilitation (venia legendi for computer science) in 2013.

Priv.-Doz. Dr.-Ing. Becker works in thefields of Petri Nets, simulation of discrete event systems and biological systems, heuristic optimization and computational intelligence. In those fields he published more than 50 journal and conference papers and served as conference co-chair, program committee member and reviewer in many conferences and journals.

**Helena Szczerbicka** is head of the Department of Simulation and Modelling - Institute of Systems Engineering at the Gottfried Wilhelm Leibniz Universität Hannover. She received her Ph.D. in Engineering and her M.S in AppliedMathematics from the Warsaw University of Technology, Poland. Dr. Szczerbicka was formerly Professor for ComputerScience at the University of Bremen, Germany. She teaches courses in discrete-event simulation, modelling methodology,queuing theory, stochastic Petri Nets, distributed simulation, computer organization and computer architecture.